

Samba & LDAP

...on Debian made simple!

mawi

Samba & LDAP: ...on Debian made simple!

mawi

Copyright © 2003

Short articles aiming to provide a concise roadmap and a stepwise test-oriented deployment tutorial of samba 2.2.x with LDAP as a backend on Debian Linux. For initiated users, yet not overly *nix technical. Does not provide background or in depth information - intended as a complement roadmap for other in depth documentation, a fairly full list of which is provided in the bibliography.

Table of Contents

Why I wrote this...	vi
Test drive!	7
OpenLDAP run-through	10
1. Basic install	11
Get, install and test OpenLDAP	11
Get, build and test Samba 2.2.x	14
2. Samba LDAP as a DC	17
Making Debian authenticate via LDAP	17
Install, configure & test smbldap-tools	18
Small FAQ	20
A. Prerequisites	22
B. The smbldap-tools config file	23
C. Config files	28
Bibliography	31
Glossary	32

List of Examples

1. Simplest samba configuration	7
1.1. Samba build rules edits	15
1.2. smb.conf edits to enable LDAP backend	16
2.1. smbldap-tools sources for /etc/apt/sources.list	18
2.2. smbldap-tools edits to smb.conf	18
2.3. smb.conf edits for samba as DC	19
4. Simple user ldif file to add to LDAP directory	20
B.1. smbldap-tools config, smbldap_conf.pm	23
C.1. base.ldif - basic directory structure	28
C.2. /etc/libnss-ldap.conf	28
C.3. /etc/pam_ldap.conf	28
C.4. /etc/pam.d/ssh	29
C.5. /etc/pam.d/passwd	29
C.6. /etc/nsswitch.conf	30

Why I wrote this...

I want to make a "cheat sheet" that includes a working configuration based on most available instructions, to get a server up quickly. A guide that just says what and not how.

I wanted to write a text that explains how to get a samba server up and running with OpenLDAP as a backend, from the perspective of primarily serving windows users.

Most other articles are lengthy, so I wanted a concise one to be available. I also wanted it to be stepwise, with tests after each step.

The cheat sheet is the goal, and not even started - I have some

Please send feedback! I really want this guide to work out - despite much testing there's always that little bug, omission or lack of clarity - mail me with your impressions/comments/errata/suggestions/"I used your article..."-mails...

First we try out samba without ldap, then uninstall it and compile LDAP and samba to make them work together, testing at the end of each section. Finally, we refine (actually redo) the LDAP directory structure and iron out wrinkles to make the solution behave as a windows user would expect, ie:

- make samba autoadd machines as we add them to the domain from the client (using the smbldap-tools scripts)
- ensure password sync
- check that the administrator group is working

The smbldap-tools are a large part of that section, largely because they've apparently become a standard way to ease administrative tasks, etc. However, they do seem to require configuring Linux authentication in order to work properly 1 which is why we will then start with configuring that end before installing the scripts.

¹I welcome any and all information about what the configuration prerequisites/requirements exactly are to get the smbldap-tools working.

Test drive!

Samba done quick - single player game!

This small article will guide you through installing default Debian Samba and testing it.



Warning

This small chapter is a waste of time! This is only to allow you to quickly test samba.

On the other hand - you should finish it in 15-30 minutes at the most!

At this point, do have a look at the pre-requisites.

Steps

1. Install the standard Debian Samba package

We use apt-get to install the package:

```
apt-get
install libcupsys2 libtiff3g samba samba-common samba-doc smbclient smbfs
```

This should both install and start the samba service.

2. Create a test share and configure samba to allow access to it

Lets create one in tmp, for example:

```
cd /tmp
mkdir test
chmod 777 test
```

We're gonna configure 2 samba using a typical test configuration (no security, etc - nothing). First we backup the original config file and then edit.

```
cd /etc/samba
mv smb.conf smb_orig.conf
nano smb.conf
```

Into the now empty config file, enter one of the simplest configurations we can use (compare by glancing at the backup config you just made):

Example 1. Simplest samba configuration

```
# -----
# Test smb.conf file
```

²Before getting ahead of ourselves, suffice it to say that samba is configured through it's config file which you can edit directly (and most changes are reflected immediately - but that differs somewhat) or use a frontend like the accompanying SWAT - which we will look at briefly below. The samba config file is in /etc/samba in Debian.

```
# mawi 2003-07-28
# -----
[global] ❶
workgroup = TESTSAMBA-GRP ❷
netbios name = TESTSAMBA ❸
security = SHARE ❹
[test] ❺
path = /tmp/test
read only = no
guest ok = yes
```

- ❶ Main section in the samba config
- ❷ As in windows - the netbios workgroup name. Doesn't matter for this test.
- ❸ The netbios name that we want this machine to have. Use the hostname of the computer for simplicity's sake.
- ❹ Interesting: The security model for the samba machine. See samba doc for more info, or hold on.
- ❺ Here we tell samba to share a directory - each general share we want is defined by it's own section. The parameters are pretty self explanatory - we indicate what folder the share is, and then that guests are ok and writing is ok.

3.

Test samba

We will make sure that samba is running, then we will test by accessing the share, then optionally test adding the samba to a domain / and or adding a client to samba (and have it act as a PDC, if only briefly). Make sure that samba is running by restarting it ;):

```
/etc/init.d/samba restart
```

List share from linux machine (locally, loopback). Try listing the samba shares using the command/program smbclient 3 and the list (L) switch:

```
smbclient -L TESTSAMBA ❶
```

- ❶ Enter the hostname of the computer (as you did in the config file above). Hopefully, you get a listing of the test share.

List and access share from a windows machine. Then try accessing the samba machine (use **net view SAM-BAHOSTNAME**) and access the share from a windows machine, as usual (using \\IPADDRESS\test). If your network 4 is working ok, everything should go fine.

Add the samba machine to an existing windows domain. Optionally, if you have a domain controller ready and administrator account to it:

```
smbpasswd -r SERVER -j RUBIES -Uadministrator%PASSWORDHERE
name and passwd of your admin acct here
```

4.

Install, enable and look at SWAT

³Samba includes not just the server programs, but also client, administration, etc programs - the whole enchilada.

⁴Instructions for getting your computers talking to each other is way out of scope for this article.

Install SWAT using apt-get:

```
apt-get  
install swat
```

Then you need to edit inet.d conf to make the web server allow access to swat using port 901 (which is the port swat uses). In Debian, the line is already there (last in the file), you just need to uncomment it to enable it, and then restart inetd (and samba just to make sure).

```
nano /etc/inetd.conf ❶  
/etc/init.d/inetd restart  
/etc/init.d/samba restart
```

❶ Edit and uncomment swat line, should be last in file
Now try the SWAT gui by surfing to it, using <http://HOSTNAME:901>. Have a look at all the tabs - pretty useful. In addition, the help links are great - now is a good time to view the help on the "security" setting in the config file. On the globals tab, find security, click help and get wiser!

5.

Uninstall samba again

This will uninstall so that we may go on to the next chapter "a-fresh" 5:

```
/etc/init.d/samba stop  
apt-get remove samba-doc samba samba-common smbclient smbfs swat winbind
```

Didn't that go quick, huh?

⁵I am not sure if this will uninstall everything, but mostly everything...

OpenLDAP run-through

OpenLDAP done quick - single player game!

This small article will guide you through compiling and testing OpenLDAP.

To be continued...

Chapter 1. Basic install

Getting ready for some multiplayer action!

Table of Contents

Get, install and test OpenLDAP	11
Get, build and test Samba 2.2.x	14

Get, install and test OpenLDAP

We will install and configure the basic slapd system so that we can test samba using it. This will omit configuring nss, pam and automating scripts and so on. This section is based on the excellent (and much more in-depth) article [amers03a], highly recommended!

Steps

1. Get source

First we create a directory to store the source in, I suggest in `/usr/local/src/`:

```
cd /usr/local/src/  
mkdir slapd slapd is the name of the server daemon program of OpenLDAP  
cd slapd/
```

Then we download the source and install dependencies to slapd, all using the Debian tool `apt-get` (if you get an error, you are probably missing the "dpkg-dev" package, see prerequisites):

```
apt-get source slapd  
apt-get build-dep slapd
```

Lastly, we install SSL development libraries (optional, if you wanna go SSL or not, but it won't hurt and only takes a sec):

```
apt-get install libssl-dev
```

2. Edit build config, build and install OpenLDAP

We will make a small adjustment to the build configuration (the "rules" 6 file), first we go into the downloaded source directory:

```
cd slapd-2.0.23/  
nano debian/rules ①
```

①the rules file is always located in the "debian" folder of a source package (directory)

- ❶ In the (rather short) "rules" file, search for "without-tls" and replace it with "with-tls". One change only.
[amers03a] says to run the rules file manually, but go with the standard dpkg-buildpackage tool (I like standard stuff):

```
dpkg-buildpackage
```

Hopefully, this goes alright. Finally we install the newly built packages using dpkg -i (for install). To do this we leave the extracted source directory:

```
cd ..
dpkg -i slapd_2.0.23-6_i386.deb \           notice backslash to split command into several
> libldap2_2.0.23-6_i386.deb \
> libldap2-dev_2.0.23-6_i386.deb \
> ldap-utils_2.0.23-6_i386.deb
```

[amers03a] says that debconf7 will ask you some questions at this point but I did not get any questions.

3.

Configure OpenLDAP, add testdata and test

We will edit the default configuration file, add two records and test the server by querying it. The configuration file is /etc/ldap/slapd.conf

```
cd /etc/ldap/
nano slapd.conf
```

The following is a simple configuration that works:

```
# /etc/ldap/slapd.conf
include /etc/ldap/schema/core.schema ❶
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/inetorgperson.schema
#include /etc/ldap/schema/nis.schema
#include /etc/ldap/schema/redhat/rfc822-MailMember.schema
#include /etc/ldap/schema/redhat/autofs.schema
#include /etc/ldap/schema/redhat/kerberosobject.schema
#include /etc/ldap/schema/samba.schema

pidfile //var/run/slapd.pid
argsfile //var/run/slapd.args

#Sample Access Control
# Allow read access of root DSE
# Allow self write access
# Allow authenticated users read access
# Allow anonymous users to authenticate
access to dn="" by * read
access to *
    by self write
    by users read
    by anonymous auth
# if no access controls are present, the default is:
# Allow read by all
# rootdn can always write

#####
##### ldbm database definitions#####
#####
database ldbm
```

⁷The Debian configuration program that may be invoked when installing new programs (packages)

```

suffix "dc=e-mf,dc=net" ❷
rootdn "cn=Manager,dc=e-mf,dc=net" ❸
rootpw SECRET ❹

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap/e-mf.net

# Indices to maintain
index primaryGroupID eq
index rid eq
index uid eq
#index uidNumber eq
#index gidNumber eq
index cn pres,sub,eq
index objectClass eq
index default sub
# ends

```

- ❶ Notice how I am at this point only leaving some schemas active (uncommented)
 - ❷ Edit the suffix
 - ❸ Enter the rootdn name. Note: all is case sensitive here!
 - ❹ Password, you can copy and use **slappasswd** crypt or md5 hash to not store in cleartext
- Let's try starting the openldap server:

```
/etc/init.d/slapd start
```

Now add a couple of records to our directory, first create a file (`base.ldif`) with the following test data:

```

dn: dc=e-mf,dc=net of course, edit this...
objectClass: domain
dc: e-mf ...and this...

dn: ou=Users,dc=e-mf,dc=net ...and this
objectClass: top
objectClass: organizationalUnit
ou: Users
description: System Users

```

Now add the data from that file using the `ldapadd` command:

```
ldapadd -x -D "cn=Manager,dc=e-mf,dc=net" -f base.ldif -W
```

(See the man page for `ldapadd` for info on the switches). You will get prompted for the password you just set. Lastly, test by querying for the records:

```
ldapsearch -x -W -D "cn=Manager;dc=e-mf;dc=net"
```

All records will be displayed. That wraps it up - OpenLDAP is basically ready to rumble!! Let's see if we can get our other contestant out in the ring!

4.

Optional but nice: Get LDAPManager and view the directory



Note

This is totally optional - but *recommended* and may please you if you feel dissapointed by the lack of *thump* in the result of our OpenLDAP installation.

There is a program called LDAPManager that will allow you to browse an LDAP directory very nicely. It is a java program and to install we need to install Java from Sun, download LDAPManager, extract and run the .jar. You then specify the server (ip) to connect to and connect using the rootdn as with ldapadd and ldapsearch above.

LDAPManager can be downloaded from its page: <http://www-unix.mcs.anl.gov/~gawor/ldap/> and <http://www.iit.edu/~gawojar/ldap/>.

Get, build and test Samba 2.2.x

This is basically a remake of the introductory article, but this time we get the sources, modify the build parameters and build Samba ourselves - with support for using LDAP as the user authentication backend 8 .

Steps

1. Download and extract source

First we create a directory to store the source in, I suggest in /usr/local/src/:

```
cd /usr/local/src/  
mkdir samba  
cd samba/
```

Then we download the source. Here you have some different options, you can use apt-get or find the source file on the net and use **wget** to download it to the directory. I'm gonna go for the latter, and I found an URL to the source (2.2.8 <http://us1.samba.org/samba/ftp/samba-2.2.8a.tar.gz>).

```
apt-get install wget install wget, if you don't have it  
wget http://us1.samba.org/samba/ftp/samba-2.2.8a.tar.gz I paste the URL I found
```

Finally, we extract the source so we can use it:

```
tar xvzf samba_2.2.8a.tar.gz skip v (verbose) if you don't want clutter
```

2. Edit build config, build and install Samba

Configuring the Samba build requires changes to some different files, so we will make a backup of these files first.

```
cd /samba-2.2.8a/debian go into the extracted source directory  
mkdir ../../ldap-deb-bak create backup directory  
cp rules libpam-smbpass.files \  
samba-common.conf files winbind.files ../../ldap-deb-bak/ then make backups
```

⁸Samba 2.x does not include support for LDAP backend in it's default package - it needs to be compiled on a machine with LDAP on it.

Three files require simple edits:

```
nano libpam-smbpass to empty
nano samba-common.conffiles to empty
nano winbind.files remove lib/security/pam winbind.so
```

The rules file (fourth and final) requires the following edits:

Example 1.1. Samba build rules edits

```
--with-pam                                remove this
--with-pam_smbpass                         remove this

--with-automount \                          add this
--with-acl-support \                       add this
--with-profile \                           add this
--disable-static \                         add this
--with-ldapsam)                            add this

131 #install -m 0644 source/nsswitch/pam_winbind.so \          comment this line
132 #$(DESTDIR)/lib/security/                  comment this line
142 #mv $(DESTDIR)/usr/bin/pam_smbpass.so $(DESTDIR)/lib/security/ comment this line
182 #cp debian/samba.pamd $(DESTDIR)/etc/pam.d/samba comment this line
```

These packages need to be installed before building, and then we go out and see the build packages and install them:

```
apt-get install libcupsys2 libtiff3g debhelper \
libcupsys2-dev autoconf libpam0g-dev libreadline4-dev
dpkg-buildpackage
cd ..
dpkg -i \
samba-common_2.2.8a-0.1_i386.deb \
libsmbclient_2.2.8a-0.1_i386.deb \
libsmbclient-dev_2.2.8a-0.1_i386.deb \
samba_2.2.8a-0.1_i386.deb \
smbclient_2.2.8a-0.1_i386.deb \
smbfs_2.2.8a-0.1_i386.deb \
swat_2.2.8a-0.1_i386.deb \
winbind_2.2.8a-0.1_i386.deb \
samba-doc_2.2.8a-0.1_all.deb
```

If all goes well, Samba should be installed and running *with* LDAP support (unconfigured).

3.

Configure and test

To test samba and start a small smb.conf, then test using it: Go through the configure and test part of the quick walk-through, then come back here and we will make some changes in order to activate the LDAP backend and test that.

Assuming that you created the simple config, lets go through the basic changes necessary to test the samba-ldap interoperability. We need to add the samba schema to the OpenLDAP config - ie copy the file to the schemas directory and then reference it in our slapd.conf: Now we need to edit the samba config, telling it to use LDAP for authentication and then we need to give samba the LDAP manager password so it can access it

for us. First we edit smb.conf:

Example 1.2. smb.conf edits to enable LDAP backend

```
;LDAP-specific settings
ldap admin dn= "cn=Manager,dc=e-mf,dc=net"           your specifics from slapd.conf
ldap port = 389
ldap ssl = no
ldap suffix = "ou=Users,dc=e-mf,dc=net"             where shall samba put/check for users?
```

Finally, we give samba the LDAP manager password, and at this point, I found restarting the services may be necessary;

```
smbpasswd -w secret your password goes here, of course
/etc/init.d/slapd restart
/etc/init.d/samba restart
```

Samba should now be able to create a user account for us in the LDAP directory based on a *nix system account (the alternative is to add the user account manually). You make a *nix account into a samba account with **smbpasswd**. So to create a samba user we add the user and then run **smbpasswd** to add the user to the directory and set the password. We will check it out using `ldapsearch` (or `LDAPmanager`) and test that we can use the user when accessing samba:

```
useradd -c "Samba User account" -d /dev/null -s /bin/false nils
ldapadd -x -h localhost -D "cn=Manager,dc=e-mf,dc=net" -f nils.ldif -W
smbpasswd -a nils
ldapsearch -x -W -D "cn=Manager;dc=e-mf;dc=net"
smbclient -L HOSTNAME -Unils%PASSWORDHERE replace here of course
```

Then try accessing the server and a share from a windows machine. If you like, to further test, try opening up SWAT and deny nils access to the share, and so on.



Further testing

Other tests at this point could be to make samba DC and test add a windows client machine. Instructions on how to do this - without LDAP considerations - can be found elsewhere.

So far so good? Next up we do a little linux changes, redo the LDAP structure and enable some scripts to throw the switch on the machine as a PDC.

Chapter 2. Samba LDAP as a DC

Table of Contents

Making Debian authenticate via LDAP	17
Install, configure & test smbldap-tools	18

Making Debian authenticate via LDAP 9

Let's change som Linux configs!

We will install and configure , PAM, passwd and ssh authentication to use LDAP. Then we will test by adding a user to the system and use ssh to log in as that user. This has nothing to do with samba except to unify the account datastore and (important) make sure smbldap-tools run.

Install:

```
apt-get install libnss-ldap libpam-ldap
```

Configure (make some backups):

```
mv /etc/libnss-ldap.conf /etc/libnss-ldap.conf_DEB-orig
nano /etc/libnss-ldap.conf example here

nano /etc/nsswitch.conf look here

nano /etc/pam_ldap.conf example here

nano /etc/libnss-ldap.conf look here for example

cp passwd passwd_DEB-orig
nano /etc/pam.d/passwd check example
```

We also need to update our ldap configuration, we need to add a nss user to the directory and in fact I advise a new structure as opposed to the quick install tests (which almost had no structure). We also edit slapd.conf to enable the nis schema:

```
nano /etc/ldap/slapd.conf uncomment nis.schema line at the beginning of the file
TODO: clear existing directory contents
ldapadd -x -D "cn=Manager,dc=e-mf,dc=net" -f base.ldif -W
Create a new base structure in a ldif file, details of my simple example here
```

Configure ssh (/etc/pam.d/ssh) to authenticate via LDAP, test:

```
cd /etc/pam.d/
cp ssh ssh_DEB-orig
nano ssh check example file
```

⁹Based on [amers03a]

To test, add a user to the directory only and try logging in using ssh with that user. Make sure the users entry in the directory has a valid shell. TODO: FIXME

Install, configure & test smbldap-tools

We need to add the sources for the Debian packages of the smbldap-tools to our apt source list file, after which we will install, configure and test. First we edit the apt sources list (/etc/apt/sources.list) and add the following lines:

Example 2.1. smbldap-tools sources for /etc/apt/sources.list

```
# smbldap-tools:
deb ftp://ftp.samba.gr.jp/pub/samba-jp/debian/woody ./
deb-src ftp://ftp.samba.gr.jp/pub/samba-jp/debian/woody ./
```

We can now install using:

```
apt-get install
smbldap-tools
```

The Debian package places the configuration file for the tools in the samba configuration directory, /etc/samba/smbldap_conf.pm, and you need to make atleast fourteen small edits. The config file is rather lengthy so I placed an annotated example and list of the edits in an appendix, so go there and have a look now! :-)

We now test that the scripts are configured correctly and working, by adding necessary groups and administrator accounts. We will restart slapd first:

```
/etc/init.d/slapd restart
TODO: restart auth or whatever?
TODO: users?
TODO: test 1 - show user
TODO: test 2 - add group
smbldap-groupadd Machines
to check if it seems to work: tail /var/log/auth.log
smbldap-groupadd -g 200 DomainAdmins Single word groupname is easiest for smb.conf, it see

smbldap-useradd -a -m -g 200 administrator
smbldap-passwd administrator

smbldap-useradd -a -m -g 200 root necessary for adding w2k/XP machines, see FAQ

smbldap-usermod -u 0 -g 0 root
smbldap-passwd root
```

We now edit smb.conf to make use of the smbldap-tools, and edit/add these lines:

Example 2.2. smbldap-tools edits to smb.conf

```
add user script = /usr/sbin/smbldap-useradd -w -d /dev/null -g Machines -c "Machine account"
password change script =
domain admin group =
```

TODO: Check and complete this!

Finally, in order to throw the switch on samba as a DC we add the following lines to smb.conf:

Example 2.3. smb.conf edits for samba as DC

TODO: FIXME!

After these last edits, restart the services and test the configuration from the windows world by: (1) Adding a computer to the domain (see FAQ for tips on how to seek out bugs in the setup), (2) add a user using the `smbldap-useradd` script/command and log in as that user, then (3) change the password for that user (4) try logging on using `ssh` (set users shell to a valid shell using `LDAPmanager`) and (5) log on as a domain administrator and test that you are getting administrative access.

Small FAQ

1. How to add a user account manually

There are several ways to add a user account (SWAT for example) - to add one manually first add it to the *nix system using either **useradd** or **adduser**, then add the user to the directory by creating a file with the data and adding it using **ldapadd** and then making it a samba account and setting its password by using **smbpasswd**.

You will want to think about whether the user shall be able to logon to the Linux machine. So assuming we have created a file with the userdata "nils.ldif":

Example 4. Simple user ldif file to add to LDAP directory

```
dn: uid=nils, ou=Users, dc=e-mf, dc=net
objectClass: inetOrgPerson
objectClass: top
objectClass: organizationalPerson
cn: Nils Nilsson test user
sn: Nilsson
title: Testuser
description: NA
```

We add this user in these three steps:

```
useradd -c "Samba User account" -d /dev/null -s /bin/false nils
ldapadd -x -h localhost -D "cn=Manager,dc=e-mf,dc=net" -f nils.ldif -W
smbpasswd -a nils
```

Notice how the user is not given a valid shell (-s) so he cannot login on the linux machine and no no valid home directory (-d).

2. How to make samba act as a PDC

TODO

3. How to add a windows client machine

Make sure "root" account exists in samba. Apparently, in order to add a win2k/XP machine, there must exist a root user in the LDAP directory. With the smbldap-tools installed we add the user like so:

```
smbldap-useradd -a -m -g 200 root Notice adding to admin group, which we have set to G
smbldap-usermod -u 0 -g 0 root TODO: hmm...? check this!
smbldap-passwd root
```

For vanilla samba (without LDAP and the smbldap-tools package) we set the password of the root account for samba using **smbpasswd**:

```
smbpasswd -a root TODO: check !
```

To add a computer to the domain, ensuring that it gets/has an account is the first step. If using LDAP and the smbldap-tools are installed and configured (and add user script of smb.conf set and working) we should be able to add a windows machine from it - remotely - so that samba adds the machine account to the LDAP directory automatically.

Otherwise, we need to add an account for the machine manually like so:

```
useradd -c "Windows NT Computer account" -d /dev/null -s /bin/false rudolf$  
  
-c = comment  
-d = home dir  
-s = shell  
  
smbpasswd -m -a rudolf create machine account in samba database
```

After the account issue has been taken care of we can set the windows client to the domain from the windows computer as usual - *except* that we need to supply the root account (and no other) when doing so. We then re-boot. All done!

Appendix A. Prerequisites

I am assuming that you know a little about linux, and especially Debian linux. For example:

- Familiarity with installing software (atleast you have used apt-get, dselect and maybe dpkg)
- Basic *nix know-how (you know how to edit a text file in linux)
- Windows network administration (you know your way around the windows "net" command)
- A test environment - do yourself a favor and get everything set for testing. You need a linux machine, a windows machine and a functioning network between them - preferably without another windows PDC on it (AFAICS). If yo have access to VMWare go for it - it really makes creating and manipulating testbeds easy! My environment was to VMWare machines on my workstation - one running linux with a bridged NIC and a host only NIC to a virtual VMWare net to which the windows machine had it's only NIC attached.
- To get sources, build, etc - you will need dpkg-dev which may not be installed on a vanilla debian system, use apt-get to get it:

```
apt-get install dpkg-dev
```

You will see me editing using nano throughout this article

Appendix B. The smbldap-tools config file

The Debian package of the tools places the configuration file in the samba configuration directory (*/etc/samba/*). This is an example of it with the fourteen necessary changes annotated. Most are fairly self explanatory. Besides these fourteen, many other parameters exist that give the scripts a necessary description of your environment.

Example B.1. smbldap-tools config, smbldap_conf.pm

```
#!/usr/bin/perl
use strict;
package smbldap_conf;

# $Id: smbldap_conf.pm,v 1.14 2002/06/01 04:30:48 olem Exp $
#
# smbldap-tools.conf : Q & D configuration file for smbldap-tools

# This code was developed by IDEALX (http://IDEALX.org/) and
# contributors (their names can be found in the CONTRIBUTORS file).
#
# Copyright (C) 2001-2002 IDEALX
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307,
# USA.

# Purpose :
# . be the configuration file for all smbldap-tools scripts

use vars qw($VERSION @ISA @EXPORT @EXPORT_OK %EXPORT_TAGS
$UID_START $GID_START $smbpasswd $slaveLDAP $masterLDAP
$with_smbpasswd $mk_ntpasswd
$ldap_path $ldap_opts $ldapssearch $ldapssearchnobind
$ldapmodify $ldappasswd $ldapadd $ldapdelete $ldapmodrdn
$suffix $usersdn $computersdn
$groupsdn $scope $binddn $bindpasswd
$slaveDN $slavePw $masterDN $masterPw
$_userLoginShell $_userHomePrefix $_userGecos
$_defaultUserGid $_defaultComputerGid
$_skeletonDir $_userSmbHome
$_userProfile $_userHomeDrive
$_userScript $usersou $computersou $groupsou
);

use Exporter;
$VERSION = 1.00;
@ISA = qw(Exporter);
@EXPORT = qw(
```

```

$UID_START $GID_START $smbpasswd $slaveLDAP $masterLDAP
$with_smbpasswd $mk_ntpasswd
$ldap_path $ldap_opts $ldapsearch $ldapsearchnobind $ldapmodify $ldappasswd
$ldapadd $ldapdelete $ldapmodrdn $suffix $usersdn
$computersdn $groupsdn $scope $binddn $bindpasswd
$slaveDN $slavePw $masterDN $masterPw
$_userLoginShell $_userHomePrefix $_userGecos
$_defaultUserGid $_defaultComputerGid $_skeletonDir
$_userSmbHome $_userProfile $_userHomeDrive $_userScript
$usersou $computersou $groupsou
);

#####
#
# General Configuration
#
#####

#
# UID and GID starting at...
#

$UID_START = 1000;
$GID_START = 1000;

#####
#
# LDAP Configuration
#
#####

# Notes: to use to dual ldap servers backend for Samba, you must patch
# Samba with the dual-head patch from IDEALX. If not using this patch
# just use the same server for slaveLDAP and masterLDAP.
#
# Slave LDAP : needed for read operations
#
# Ex: $slaveLDAP = "127.0.0.1";
$slaveLDAP = "127.0.0.1";

#
# Master LDAP : needed for write operations
#
# Ex: $masterLDAP = "127.0.0.1";
$masterLDAP = "127.0.0.1";

#
# LDAP Suffix
#
# Ex: $suffix = "dc=IDEALX,dc=ORG";
$suffix = "dc=e-mf,dc=net"; ❶

#
# Where are stored Users
#
# Ex: $usersdn = "ou=Users,$suffix"; for ou=Users,dc=IDEALX,dc=ORG
$usersou = q(_USERS_); ❷

$usersdn = "ou=$usersou,$suffix";

#
# Where are stored Computers

```

```
#
# Ex: $computersdn = "ou=Computers,$suffix"; for ou=Computers,dc=IDEALX,dc=ORG
$computersou = q(_COMPUTERS_); ❸

$computersdn = "ou=$computersou,$suffix";

#
# Where are stored Groups
#
# Ex $groupsdn = "ou=Groups,$suffix"; for ou=Groups,dc=IDEALX,dc=ORG
$groupsou = q(_GROUPS_); ❹

$groupsdn = "ou=$groupsou,$suffix";

#
# Default scope Used
#
$scope = "sub";

#
# Credential Configuration
#
# Bind DN used
# Ex: $binddn = "cn=Manager,$suffix"; for cn=Manager,dc=IDEALX,dc=org
$binddn = "Manager"; ❺

#
# Bind DN passwd used
# Ex: $bindpasswd = 'secret'; for 'secret'
$bindpasswd = "asda1452"; ❻

#
# Notes: if using dual ldap patch, you can specify to different configuration
# By default, we will use the same DN (so it will work for standard Samba
# release)
#
$slaveDN = $binddn;
$slavePw = $bindpasswd;
$masterDN = $binddn;
$masterPw = $bindpasswd;

#####
#
# Unix Accounts Configuration
#
#####

# Login defs
#
# Default Login Shell
#
# Ex: $_userLoginShell = q(/bin/bash);
$_userLoginShell = q(_LOGINSHELL_); ❷

#
# Home directory prefix (without username)
#
#Ex: $_userHomePrefix = q(/home/);
$_userHomePrefix = q(/home/);
#
# Gecos
#
$_userGecos = q(System User);
```

```

#
# Default User (POSIX and Samba) GID
#
$_defaultUserGid = 100; 8

#
# Default Computer (Samba) GID
#
$_defaultComputerGid = 553; 9

#
# Skel dir
#
$_skeletonDir = q(/etc/skel); 10

#####
#
# SAMBA Configuration
#
#####

#
# The UNC path to home drives location without the username last extension
# (will be dynamically prepended)
# Ex: q(\\My-PDC-netbios-name\homes) for \\My-PDC-netbios-name\homes
$_userSmbHome = q(\\_PDCNAME_\homes); 11

#
# The UNC path to profiles locations without the username last extension
# (will be dynamically prepended)
# Ex: q(\\My-PDC-netbios-name\profiles) for \\My-PDC-netbios-name\profiles
$_userProfile = q(\\_PDCNAME_\profiles\); 12

#
# The default Home Drive Letter mapping
# (will be automatically mapped at logon time if home directory exist)
# Ex: q(U:) for U:
$_userHomeDrive = q(_HOMEDRIVE_); 13

#
# The default user netlogon script name
# if not used, will be automatically username.cmd
#
#$_userScript = q(startup.cmd); # make sure script file is edited under dos 14

#####
#
# SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
#
#####

# Allows not to use smbpasswd (if $with_smbpasswd == 0 in smbldap_conf.pm) but
# prefer mkntpwd... most of the time, it's a wise choice :-))
$with_smbpasswd = 0;
$smbpasswd = "/usr/bin/smbpasswd";
$mk_ntpasswd = "/usr/local/sbin/mkntpwd";

$ldap_path = "/usr/bin";
$ldap_opts = "-x";
$ldapsearch = "$ldap_path/ldapsearch $ldap_opts -h $slaveLDAP -D '$slaveDN' -w '$slavePw'";
$ldapsearchnobind = "$ldap_path/ldapsearch $ldap_opts -h $slaveLDAP";
$ldapmodify = "$ldap_path/ldapmodify $ldap_opts -h $masterLDAP -D '$masterDN' -w '$masterP'";
$ldappasswd = "$ldap_path/ldappasswd $ldap_opts -h $masterLDAP -D '$masterDN' -w '$masterP'";

```

```
$ldapadd = "$ldap_path/ldapadd $ldap_opts -h $masterLDAP -D '$masterDN' -w '$masterPw'";  
$ldapdelete = "$ldap_path/ldapdelete $ldap_opts -h $masterLDAP -D '$masterDN' -w '$masterPw'";  
$ldapmodrdn = "$ldap_path/ldapmodrdn $ldap_opts -h $masterLDAP -D '$masterDN' -w '$masterPw'";  
  
1;  
  
# - The End
```

- ❶ Root suffices of LDAP (as in slapd.conf)
- ❷ Name of LDAP directory holding all users
- ❸ Name of LDAP directory to place computer accounts.
- ❹ Name of the LDAP directory holding all domain groups
- ❺ The name of the rootdn to use to bind to the LDAP directory
- ❻ The password of the rootDN as you've set in slapd.conf
- ❼ Default shell of all new created accounts
- ❽ Default Group ID of all users created
- ❾ Default Group ID (GID) of computers to be given by the scripts
- ❿ Location of home directory skeleton for all new users
- ⓫ Sharename of home
- ⓬ Sharename of profile shares
- ⓭ Driveletter for home directory
- ⓮ Name of script for all users

Appendix C. Config files

These are the final versions of the configuration files mentioned in my articles. They do not represent final-optimized versions - they are simplistic and minimal versions in the spirit of the article, provided here in one place - far from a production grade configuration.

Example C.1. base.ldif - basic directory structure

```
TODO: FIX ME
```

This base.ldif does not include groups since I create them using the smbldap-tools

Example C.2. /etc/libnss-ldap.conf

```
##### /etc/libnss-ldap.conf #####
host localhost
base ou=Users,dc=e-mf,dc=net
uri ldap://10.0.0.33 host address
ldap_version 3

binddn cn=nss,dc=e-mf,dc=net the nss user you added previously
bindpw qwerty

#nss_base_passwd ou=Users,dc=e-mf,dc=net
nss_base_passwd dc=e-mf,dc=net root suffix - since we have accounts in two places (?)
nss_base_group ou=Groups,dc=e-mf,dc=net where to store groups
#####
```

```
src: [amers03a]
```

Example C.3. /etc/pam_ldap.conf

```
##### /etc/pam_ldap.conf #####
# http://homex.subnet.at/~max/ldap/
#
# pam_ldap.conf for all client machines

host virtdeb enter hostname, base LDAP surfix, host address...
base dc=e-mf,dc=net
uri ldap://10.0.0.33/
ldap_version 3

rootbinddn cn=Manager,dc=e-mf,dc=net ...and rootdn to access

pam_password crypt
#####
```

src: [amers03a]

Example C.4. /etc/pam.d/ssh

```

#%PAM-1.0
auth      required      pam_env.so
#auth     required      pam_nologin.so
auth      sufficient    pam_ldap.so
auth      required      pam_unix.so

account   sufficient    pam_ldap.so
account   required      pam_unix.so

session   sufficient    pam_ldap.so
session   required      pam_unix.so
session   optional      pam_lastlog.so # [1]
session   optional      pam_motd.so # [1]
session   optional      pam_mail.so standard noenv # [1]
session   required      pam_limits.so

password  sufficient    pam_ldap.so
password  required      pam_unix.so

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
#
# password required      pam_cracklib.so retry=3 minlen=6 difok=3
# password required      pam_unix.so use_authtok nullok md5

```

Note the changes to original. src: [amers03a]

Example C.5. /etc/pam.d/passwd

```

#
# The PAM configuration file for the Shadow `passwd' service
#
# The standard Unix authentication modules, used with NIS (man nsswitch) as
# well as normal /etc/passwd and /etc/shadow entries. For the login service,
# this is only used when the password expires and must be changed, so make
# sure this one and the one in /etc/pam.d/login are the same. The "nullok"
# option allows users to change an empty password, else empty passwords are
# treated as locked accounts.
#
# (Add `md5' after the module name to enable MD5 passwords the same way that
# `MD5_CRYPT_ENAB' would do under login.defs).
#
# The "obscure" option replaces the old `OBSOURE_CHECKS_ENAB' option in
# login.defs. Also the "min" and "max" options enforce the length of the
# new password.

```

```
password    sufficient    pam_ldap.so
password    required          pam_unix.so nullok obscure min=4 max=8 md5

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSOLETE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
# password required          pam_cracklib.so retry=3 minlen=6 difok=3
# password required          pam_unix.so use_authtok nullok md5
```

Note the changes to original. src: [amers03a]

Example C.6. /etc/nsswitch.conf

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:      ldap compat
group:       ldap compat
shadow:      ldap compat

hosts:       files dns
networks:    files

protocols:   db files
services:    db files
ethers:      db files
rpc:         db files

netgroup:    nis
```

Note the changes to original. src: [amers03a]

Bibliography

Top reads!

[amers03a] *Using OpenLDAP on Debian Woody to serve Linux and Samba users*. Markus Amersdorfer. August 2003. URL: [http://homex.subnet.at/~max/ldap/ Recommended!](http://homex.subnet.at/~max/ldap/Recommended!) If you don't like me, this is where you should go. Best of all writings I've looked at..

[amers03b] *How to join a Windows client to a domain*. Markus Amersdorfer. August 2003. http://homex.subnet.at/~max/comp-06_samba-pdc.shtml .

[idealx1] *Samba LDAP How to*. Olivier Lemaire. IDEALX S.A.S. URL: <http://www.idealx.org/prj/samba/samba-ldap-howto.pdf> If not the first than one of the first samba LDAP how-to's. Almost all other articles have some info from this one. (Project info: <http://www.idealx.org/prj/samba/index.en.html> .

[howtono3] David Trask. *The latest SAMBA-LDAP-PDC How-to (another one)*. URL (PDF): http://www.vcs.u52.k12.me.us/LDAP/The_SAMBA-LDAP_How-to.pdf URL (HTML): http://www.vcs.u52.k12.me.us/LDAP/The_SAMBA-LDAP_How-to.html The only simple info on editing the idealx scripts config file, smbldap_conf.pm..

URL: <http://www.mandrakesecure.net/en/docs/samba-pdc.php> . Mandrake.

URL: <http://network.gouldacademy.org/randomfiles/sambaldap/SambaLDAP/index.html> Nice and concise, some info pieces here that are no where else - on the idealx scripts..

Runners up!

[icoup03] *SAMBA (v 2.2) PDC LDAP v.3 howto (unofficial)*. Ignacio Coupeau. May 2003. University of Navarra. URL: http://www.unav.es/cti/ldap-smb/ldap-smb-2_2-howto.html Mr Coupeau obviously has alot of experience and previously this was one of the few sources of info available (I think). Now there is alot more available and even though some valuable info is available here and no where else, it is crudely written and unstructured..

Using an LDAP Directory for Samba Authentication. Tom Syroid. IBM Developerworks. URL: http://www.ibm.com/servers/esdd/tutorials/smb_ldap/smb_ldap-ltr.pdf Not bad but like most stuff very Red Hat focused, I started out using this because it is well written and simple. Mr Syroid also has a good article on samba as a PDC (no LDAP)..

Other references

Samba LDAP Debian How-to. URL: <http://howto.aphroland.de/HOWTO/LDAP/FrontPage> Nice but technical. Amersdorfer seems to have used this one to some extent..

URL: <http://www-106.ibm.com/developerworks/linux/library/l-samba/> .

LDAP System Administration. O'Reilly. 1-56592-491-6. Gerald Carter. Good for explaining LDAP, OpenLDAP, Linux authentication, etc. Not much on samba though..

Glossary

Some terms used

NET command (windows)

Actually *many* commands to view, test, administer and manipulate windows network access from a windows machine. Examples include: net view, net use, net print, etc.

Name Service Switch

handles mapping between names and numbers, dealing with groups and access. See [nakedape](http://nakedape.cc/wiki/index.cgi/NameServiceSwitch) [wikipedia](http://wikipedia.org/wiki/NameServiceSwitch) [http://nakedape.cc/wiki/index.cgi/NameServiceSwitch].

Pluggable Authentication Module

handles authentication for access to different resources on the machine. Like so many things in Linux, you configure it a little differently between distros (because they include different options).

Samba Web Administration Tool (something)

A web based administration program for samba, installed by default - but not enabled by default!

Samba configuration file

The samba configuration file, in Debian located in `/etc/samba/`. The default is nicely documented, although I recommend glancing at it using SWAT - and clicking "advanced view", then use the help links to read up on interesting looking parameters.

As Far As I Can See

Extremely silly but useful and after a while more so (addictive) typical newsgroup and netspeak short.